MENTAL HEALTH IN TECH SURVEY ANALYSIS CODE DESCRIPTION

Introduction:

This document provides an explanation of the Python code used to analyze the "Mental Health in Tech Survey" dataset, incorporating detailed explanations of the statistical concepts, coding theory, nuances, and specific techniques employed.

The survey dataset is primarily focused on mental health in the tech sector, exploring how factors like company policies, demographic information, and job characteristics relate to seeking mental health treatment and outcomes. Analyzing this data can help identify factors that either support or inhibit mental health care within tech workplaces.

Theoretical Considerations

1. **Objective:** Identify whether certain workplace factors and demographics (e.g., whether a company's wellness program discussion includes mental health, gender, family history of mental health issues) correlate with mental health support and treatment-seeking behavior.
2. **Hypotheses:** The analysis is based upon the following hypotheses:
    - When a company's wellness program discussion includes mental health, the company has higher rates of employees seeking mental health treatment.
    - Demographics, including gender and family history, are associated with different levels of mental health treatment-seeking behavior.
    - Cultural and regional differences impact workplace mental health support.

Methodological Justification

The notebook applies a range of statistical and machine learning techniques, each selected to meet specific analytical goals:

1. **Descriptive Statistics and Visualization:** Help understand distributions and categorical relationships.
2. **Feature Engineering and Selection:** Uses statistical tests (Chi-square) to select features, refining predictive models.
3. **Predictive Modeling (Logistic Regression):** Tests the relationships between features and outcomes.

Code Summary:

1. Data Preprocessing: Loading and Initial Cleaning:

- **Libraries:** A detailed breakdown of the imported libraries:

- os: Provides functions for interacting with the operating system, crucial for tasks like checking file existence and running system commands (e.g., for downloading the Kaggle dataset).
- zipfile: Specifically used for working with zip archives. This is included to handle the downloaded dataset from Kaggle, which is often packaged as a zip file.
- pandas: The core library for data manipulation and analysis. It provides the DataFrame structure, which is used throughout the code.
- numpy: Provides efficient numerical operations, array manipulation, and mathematical functions, which are essential for statistical calculations and data preprocessing.
- seaborn: Built on top of matplotlib, seaborn provides a high-level interface for creating informative and statistically relevant visualizations. Its functions often handle the complexities of creating statistically sound plots, like handling different plot types and statistical estimations.
- matplotlib.pyplot: The fundamental plotting library in Python. Seaborn uses matplotlib under the hood, but matplotlib can be used directly for more fine-grained control over plots.
- statsmodels: Provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests and statistical data exploration. It offers more statistically-focused functionalities than scikit-learn.
- sklearn (scikit-learn): A powerful machine learning library with tools for data preprocessing, feature selection, model building, and evaluation. Its SelectKBest and LabelEncoder are used here.

- **Data Acquisition and Verification:** The code includes a robust mechanism to acquire the dataset. First, it checks if the zipped dataset file already exists locally. If not, it leverages the Kaggle API via os.system to download the dataset directly. This approach promotes reproducibility by ensuring everyone starts with the same data, regardless of whether they already have the file. After downloading, the code extracts the CSV from the zip archive using zipfile.ZipFile. This automated process streamlines data acquisition.
- **Missing Data Imputation Strategies:**
  - **Column-wise Deletion:** The code removes columns with a high percentage of missing values (greater than 90%). This addresses the issue where imputing values in such columns could introduce substantial bias due to the large proportion of missing data. The threshold is calculated dynamically (0.9 * len(df)), which adapts to different dataset sizes.
    - **Statistical Justification:** Removing columns with over 90% missing values reduces dimensionality without substantial loss of information. It is generally

acceptable to drop variables with high missing rates, especially when the data cannot be reasonably imputed.

    - **Row-wise Deletion:** Rows with missing values in the 'tech company' column are removed. This targeted deletion is justified because the analysis focuses on tech employees, and imputing these values could skew the results.
    - **Categorical Imputation:** For specific categorical columns ('self-employed', 'state', 'work interfere'), missing values are imputed with "Other". This strategy preserves the distribution of existing categories and prevents the introduction of new, potentially meaningless categories. This approach is preferred over simply dropping the rows because it retains valuable data points.

- **Age Outlier Management:** The code replaces age outliers (values outside the realistic range of 18-75) with the mean age within the valid range. This method is chosen over deletion to preserve sample size, while mitigating the influence of potentially erroneous age entries. Calculating the mean dynamically using boolean indexing df[(df['Age'] >= 18) & (df['Age'] <= 75)]['Age'].mean() is a robust and efficient way to find the mean of the valid data.
    - **Theoretical Justification:** Age outliers are replaced with the mean to prevent potential distortions in analysis. It also reflects a common approach in applied data analysis to maintain realistic boundaries (e.g., setting a working age range).

2. Data Preprocessing and Feature Engineering:

- **Gender Standardization:** The code uses a dictionary (col_renames) to store mappings for renaming columns. This is particularly efficient when dealing with multiple column renames. Regular expressions could be a more powerful approach for complex pattern matching during standardization, but the list-based approach with replace is sufficient for this specific task.
    - **Purpose:** Simplifying gender categories reduces complexity and clarifies analysis. Transforming responses into consistent categories (Male, Female, Other) enables easier interpretation and supports accurate gender-based analysis.
    - **Statistical Consideration:** Simplified categories may generalize or obscure nuanced responses (e.g., non-binary identities), which could affect findings related to gender diversity.
- **Benefits Standardization:** Standardizing 'benefits' ensures consistent analysis. This process improves data quality by grouping synonymous responses.
- **Categorical Feature Encoding - Label Encoding vs. One-Hot Encoding:** Label encoding assigns a unique integer to each category within a feature. This is efficient in terms of memory and computational cost compared to one-hot encoding, which creates new dummy variables for each category. One-hot encoding can lead to high dimensionality, especially with features having many unique values, which can impact model performance and interpretability, especially for algorithms like Logistic Regression that can be prone to multicollinearity. Label encoding, however, introduces an ordinal relationship between

categories, which might not be inherently present in the data. This trade-off is considered, and label encoding is deemed suitable in this specific analysis.

3. Exploratory Data Analysis (EDA) and Visualization:

- **Visualizations - Best Practices:**
    - **Color Palettes:** Using sns.color_palette("pastel") provides visually appealing and consistent color schemes across different visualizations.
    - **Titles and Labels:** Clear and concise titles (plt.title) and axis labels (plt.xlabel, plt.ylabel) are essential for communicating insights effectively.
    - **Annotations:** Annotations on bar plots and histograms, directly displaying counts or percentages, enhance readability and understanding. Dynamically positioning annotations based on bar positions and heights ensures accurate placement.
    - **Legends:** Legends (plt.legend) are crucial for interpreting plots with multiple categories or groups. Customized legends using Line2D allow for greater control over legend appearance and content, enabling the inclusion of percentages or other relevant information.
- **FacetGrid - Advanced Usage and Customization:** The modified_hist function demonstrates the power of customizing plots within a FacetGrid. Calculating percentages within bins and dynamically placing annotations directly on the histograms provides a richer visual representation of the distributions within different subgroups.
- **Correlation Matrix - Statistical Interpretation:** The correlation matrix provides valuable insights into the linear relationships between numerical features. Identifying highly correlated pairs is important for understanding potential multicollinearity issues, which can affect the stability and interpretability of regression models. A correlation threshold of 0.33 is used to identify potentially problematic correlations. This value can be adjusted based on the specific domain and analytical goals.
- **Linear Regression - Assumptions and Interpretations:** Linear regression assumes a linear relationship between variables, homoscedasticity (constant variance of residuals), independence of errors, and normally distributed residuals. The R-squared value measures the proportion of variance in the dependent variable explained by the independent variable. The code incorporates a check for constant columns to avoid errors and skips plots where the R-squared is below a threshold (0.12 by default), indicating a weak linear relationship.
- **Select Visualizations:**
    - **Gender Distribution and Family History**
        - **Interpretation:** The plot provides a visual comparison of family history presence across genders. By examining counts within each gender category, the analysis investigates any potential gender-based predisposition to mental health issues.
        - **Statistical Consideration:** Family history could act as a confounding variable. Its presence or absence may influence both mental health treatment-seeking

behavior and other variables (e.g., workplace environment or personal attitudes toward mental health).

- **Employee Expectation that Discussing Mental Health with Employer Will Have Negative Consequences and Gender**
  - **Purpose:** Counts the respondent expectation that discussing mental health with employer will have negative consequences across genders. Differences might imply varying mental health impacts or cultural influences affecting mental health among different genders.
- **Treatment-Seeking Behavior Rates by Whether a Company's Wellness Program Discussion Includes Mental Health**
  - **Interpretation:** The count plot shows treatment-seeking frequencies by whether a company's wellness program discussion includes mental health. This is aimed at determining what, if any, role exists for including mental health in a company's wellness program discussion with regard to encouraging or supporting mental health care.
  - **Statistical Analysis:** This visualization could be complemented by calculating relative risk or odds ratios to quantify how much more likely treatment-seeking behavior is for those respondents who work for an employer where wellness program discussions include mental health.
- **Ease of Taking Leave for Mental Health by Country**
  - **Interpretation:** This bar plot (with error bars) compares ease of taking leave for mental health across countries. Higher scores suggest greater ease of taking leave for mental health, which could relate to mental health treatment-seeking behavior by employees.
  - **Statistical Consideration:** An analysis of variance (ANOVA) could be conducted to test if the mean ease of taking leave for mental health scores significantly differ across countries.
- **When a Company's Wellness Program Discussion Includes Mental Health:** Analyzing treatment-seeking behavior rates with regard to whether a company's wellness program discussion includes mental health provides insights into the impact of when a company's wellness program discussion includes mental health on treatment-seeking behavior rates of the company's employees. The code displays both counts and percentages, allowing for a more comprehensive understanding of the relationship. Statistical tests (e.g., chi-squared test) could be used to assess the statistical significance of the differences in treatment-seeking behavior rates between program types.
- **Company Size Analysis:** Similar to the analysis of when a company's wellness program discussion includes mental health, the code investigates the relationship between company size and treatment-seeking behavior. Statistical tests and effect size measures can provide further insights into the strength of the relationship.
- **Country-Specific Analysis:** This analysis allows for comparisons of mental health outcomes and workplace factors across different countries. It's important to consider potential cultural and societal factors that may influence these outcomes.
  - **Defining the Scoring System**
  - **Calculating Weighted Mean Scores**

- **Deriving Mean Scores and Standard Deviations for Countries**
- **Consolidating Results into DataFrames**
- **Visual Representation of Mean Scores**
- **Displaying Tabular Results**

4. Feature Selection - Statistical Significance:

- **Chi-squared Test - Rationale and Limitations:** The chi-squared test is used for feature selection with categorical features because it assesses the statistical significance of the association between features and the target variable ('treatment'). Features with higher chi-squared scores are considered more relevant. The test assumes that the categories are mutually exclusive and that the expected cell counts are not too small (typically > 5). While effective for categorical features, the chi-squared test doesn't account for interactions between features.
    - **Statistical Theory:** Chi-square tests assess whether observed frequencies differ from expected frequencies under the assumption of independence between feature and outcome. Low p-values suggest that the feature is informative for predicting the outcome.
    - **Applicability:** Chi-square is suitable for nominal or ordinal data but may produce misleading results if applied to continuous variables without discretization.
    - **Selection Threshold:** Instead of selecting all features (`k='all'`), the code could set a threshold for Chi-square scores, keeping only the most predictive features. This would further refine the model's focus on relevant variables.

5. Predictive Modeling - Model Evaluation and Interpretation:

- **Logistic Regression:** Logistic regression is suitable for binary classification (predicting 'treatment' as 'Yes' or 'No'). The code utilizes L1 regularization (Lasso) for feature selection and regularization, which is helpful when dealing with potentially many correlated features. The liblinear solver is efficient for moderately sized datasets. Model evaluation metrics like accuracy, precision, recall, F1-score, and AUC should be used to assess model performance.
    - **Statistical Theory:** Lasso (L1 regularization) introduces a penalty term that shrinks some coefficients to zero, effectively performing feature selection. Lasso's sparsity-inducing property helps retain only the most predictive features, reducing overfitting and enhancing model interpretability.
    - **Interpretation:** After fitting, the model's non-zero coefficients indicate influential predictors of mental health treatment-seeking behavior.
    - **Methodological Justification:** Using L1 regularization in logistic regression is appropriate for datasets with many features, as it simplifies the model by excluding non-contributory variables.
- **Regression Output Interpretation:** The results.summary() output from statsmodels provides information about model fit (log-likelihood, AIC, BIC), coefficient estimates (magnitude and

p-values), and overall model significance. Interpreting p-values and confidence intervals is crucial for understanding the statistical significance of the relationships between predictors and the outcome variable.

- **Purpose:** Provides a full regression model without regularization, producing statistical significance (p-values) and confidence intervals for each predictor.
- **Interpretation:** The output summary table includes coefficients and p-values, allowing assessment of each feature's contribution to the likelihood of mental health treatment-seeking behavior.
- **Considerations:** Logistic regression without regularization may result in multicollinearity, especially in high-dimensional datasets, which can inflate coefficient variance.